

# Python & Java {4} Teachers



# The Caesar Cipher

Level 1



Funded by the  
Erasmus+ Programme  
of the European Union

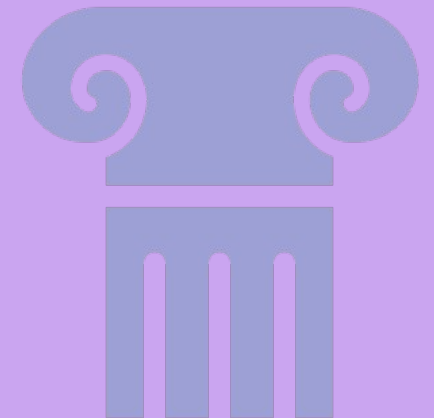
{4}

Python & Java 4 Teachers

# Introduction

At work- At home-  
Helping others

Caesar commanded all of Rome's armies and won many battles that gave more land to Rome. How was he so successful as a leader? The key was his communication with his armies, which was encrypted so that his enemies could not decipher his battle plans

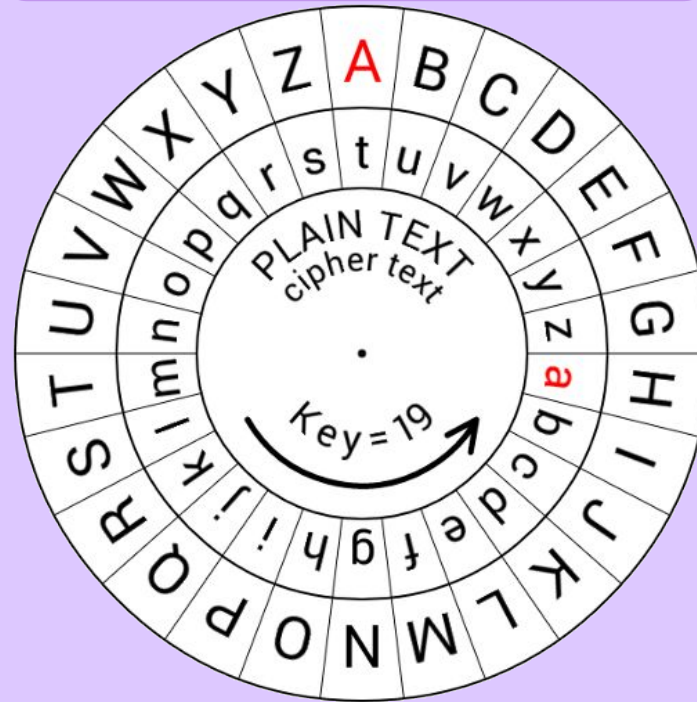


# The Caesar Cipher

A cipher is a type of secret code, where you swap the letters around so that no-one can read your message. Hiding a word is called *encryption*.

The Caesar cipher works by moving each letter in the plain text (uncoded message) up or down by a certain number of places in the alphabet (determined by the key). Can be modelled using a wheel.

The inner wheel will represent the new letter and the outer wheel will represent the original text.



On this example, the key is 19, meaning that the inner circle has been moved 19 times to the right



# Encoding a message...

Using the same wheel as the previous slide we will code the message

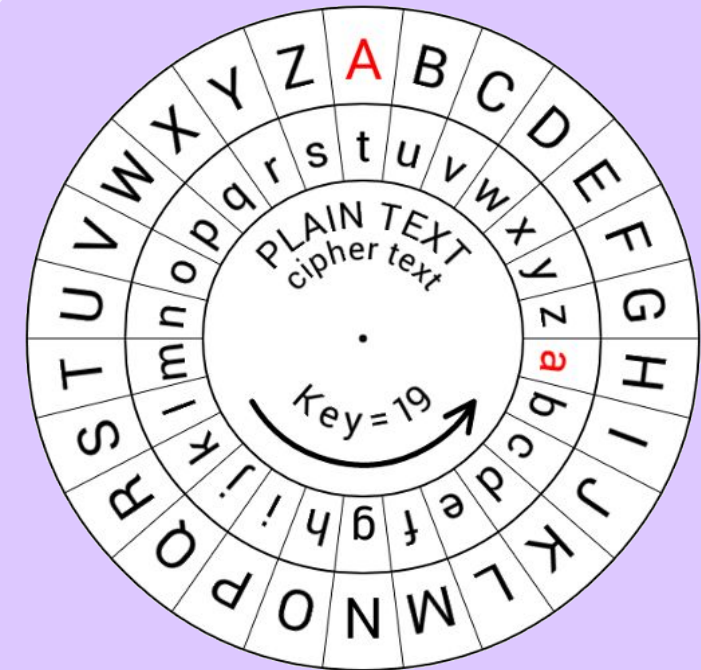
“PYTHON PROJECT”

Step 1: what is the key? (in this case 19)

Step 2: move the inner wheel that many places to the right (or left if the key is negative)

Step 3: now that the inner and outer wheel are lined up, for each letter of the plain text (“PYTHON PROJECT”), find the encoded equivalent (so “P” will change to “i”)

Step 4: do this for every letter



Plain text: PYTHON PROJECT

Encoded message: irmahg ikhcxvm



# Task

- Create a program to encode messages using the Caesar Cipher





# Process

This program should:

- ✓ Ask the user for a piece of plain text to be encrypted
- ✓ Ask the user for a key
- ✓ Encrypt the plain text using the key
- ✓ Output the encrypted message to the user



# Input and Output

```
print("Hello World!")
```

Input is the process of receiving information from the user.

We do this using the `input()` statement.

We can store the user's data in variables!

Output is the process of supplying information to the user.

In python, we use the `print()` statement.

```
print("What is your name?: ")  
name = input()
```



# Step 1

## Introducing the program

Here, the print statement is used to display to the user, in this case, the title of the project

```
print("Welcome to the Caesar Cipher software!")
```



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



# Data types

In programming, every variable will be given a data type (information to the computer about what type of data is stored in that variable). On the next slide, you will be introduced to two new data types: **string** and **integer**

String	Integer
<ul style="list-style-type: none"><li>• String is the default data type in python</li><li>• We store text in strings</li><li>• We can access different positions in the string using square brackets []</li></ul>	<ul style="list-style-type: none"><li>• Integer is the data type that stores whole numbers (for example: 1,2,3,4)</li><li>• Because string is the default, if we want an integer, we have to manually change it using <code>int()</code></li></ul>
<pre>alphabet = "abcdefghijklmnopqrstuvwxyz"</pre>	<pre>key = int(input("Enter the key: "))</pre>

# Step 2

Declaring all variables

```
alphabet = "abcdefghijklmnopqrstuvwxyz"
```

The variable alphabet is a string (which is a variable composed of text)

```
newMessage = ""
```

The variable newMessage is saved as an empty string (just "") and will be added to later in the program

```
key = int(input("Enter the key: "))
```

Integer data type defined here using int()

The input statement allows the user to input the key that they want to use (remember: the key is how many positions along the alphabet the encoded text will be)

```
message = input("Please enter a message: ")
```



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



# Loops

A loop is a sequence of instructions that is continually repeated until a certain condition is reached.

In Python there are two main loops: 'FOR Loops' (we only need for loops for this program) and 'WHILE Loops'

While Loops are condition controlled and will repeat until their condition is false.

```
1
2 condition = True
3 while condition:
4     print("Repeating...")
5
6     print("Finish loop?")
7     finished = input()
8
9
10    if finished == "Y":
11        condition = False
12
```

```
Repeating...
Finish loop?
N
Repeating...
Finish loop?
N
Repeating...
Finish loop?
N
Repeating...
Finish loop?
Y
```

For loops are count controlled and will repeat a set number of times.

```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

# Step 3

For loop and if statement

```
for character in message:
```

In this program, for the for loop, instead of using “for i in range:” we will be using “for character in message:”. This means that it will look at the variable “message” that has been entered by the user, and repeat the code for the amount of characters in the message- in the example “PYTHON PROJECT” there are 14 characters (including spaces)



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



# If statements

```
if character in alphabet:  
    print ("the character is in the alphabet")  
else:  
    print ("the character is NOT in the alphabet")
```

The IF statement is a decision-making statement that guides a program to make decisions based on specified criteria.

The IF statement executes one set of code if a specified condition is met (TRUE) or another set of code evaluates to FALSE.



# Step 4

## If statement

```
for character in message:  
    if character in alphabet:  
  
    else: ←  
        newMessage += character
```

An example of this is if a symbol like one of @f\*() is used

It is a similar principle for the if statement here as can be seen. Depending on which iteration of the for loop the program is on (the number of how many times the code has run for) depends on which character is looked at. For example, the 4<sup>th</sup> time the for loop is run, the following code will look at the 4<sup>th</sup> character in the variable “message”. It will then see if that letter is contained in the variable alphabet, and if not, then the plaintext cannot be encoded, and the plaintext character will just be added to the string newMessage (declared in step 1) instead



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



# String methods

`find():`

- Returns the information stored at the string's index location that is put in the brackets afterwards

```
position = alphabet.find(character)
```

Accessing strings (will be used in step 6):

- Individual characters can be accessed using indexing with square brackets
- In programming, the computer starts counting from 0 so to get the first character in a string, it will be at position [0]- in the example "PYTHON PROJECT":
  - [0] is P
  - [1] is Y
  - [2] is H
  - E.c.t

# Step 5

String methods- find ()

```
if character in alphabet:  
    position = alphabet.find(character)
```

In this line of code, the program will create a variable called position and store in it whatever is stored in the index of where the loop is at in its course of running.

For example, the 5<sup>th</sup> time the for loop is run, the following code will look at the 5<sup>th</sup> character in the variable “message” and store it in the variable “position”. In the example “PYTHON PROJECT”, the 5<sup>th</sup> character is N



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



# Basic arithmetic operators

Addition, subtraction, multiplication and division are all very easy to do in Python! However, the operators used to perform each function are slightly different than the ones we use in mathematics.

```
num1 = 15  
num2 = 3
```

```
print(num1 + num2)  
print(num1 - num2)  
print(num1 * num2)  
print(num1 / num2)
```

```
18  
12  
45  
5.0
```

Instead of 'X' and '÷' to multiply and divide, we use \* and /.

# Harder arithmetic operators

In python, you can use the operator `+=` to increase something by a certain amount (in the example 1 is added to num1 (15+1=16))

```
num1 = 15
num2 = 3
num3 = 19

num1 += 1
print(num1)
print(num3 % num2)
```

```
16
1
```

When dividing two numbers that don't divide exactly (like 19 and 3), there will be a remainder. In the example:

$$19 / 3 = 6$$

Here, there will be a 1 left over, called the remainder. If we were to instead do the operation:

$$19 \% 3 = 1$$

# Step 6

Accessing strings [] and harder arithmetic operators

```
newPosition = (position + key) % 26
```

Here, the variable “newPosition”, is equal to the remainder (%) of the sum of the position (explained in step 5) and the key, when divided by 26. For example, if the plain text character was a C (in position 2 in the alphabet array due to the zero index in python) , and the key was 4. The newPosition=(2+4) % 26 which is 6

```
newCharacter = alphabet[newPosition]
```

Here, the variable “newCharacter”, is equal to the “newPosition”. For example, if like above, the newPosition is equal to 6, it will correlates to the character g in the alphabet list

```
newMessage += newCharacter
```

Here, the variable “newMessage” will have the new character added to it. Without this line, instead of forming the whole message (for example “PYTHON PROJECT”), every time the loop runs it will not save the new character that has been encrypted

```
newCharacter = alphabet[newPosition]
```



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers





# What the code should look like...

```
1 print("Welcome to the Caesar Cipher software!")
2
3 alphabet = "abcdefghijklmnopqrstuvwxyz"
4 newMessage = ""
5 key = int(input("Enter the key: "))
6 message = input("Please enter a message: ")
7
8 for character in message:
9     if character in alphabet:
10         position = alphabet.find(character)
11         newPosition = (position + key) % 26
12         newCharacter = alphabet[newPosition]
13         newMessage += newCharacter
14     else:
15         newMessage += character
16
17 print("The new message is: " + newMessage)
```



That's a lot more complicated than the way I did it with the wheel ... but brilliant well done!

```
Welcome to the Caesar Cipher software!
Enter the key: 4
Please enter a message: you are finished !
The new message is: csy evi jmmwlih !
>>> |
```

# Extension...

{4}

Python & Java 4 Teachers

See if you can do this in reverse to decrypt the code! Otherwise how will your correspondent see the message?





# Extension solution...

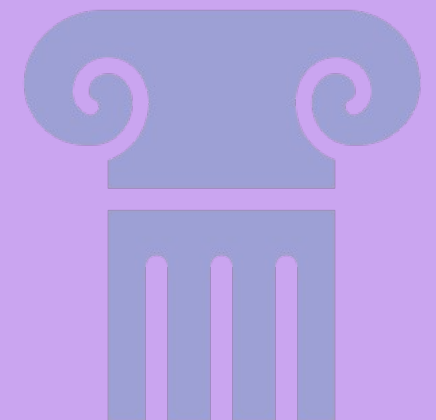
```
restoredMessage = ""  
  
key = int(input("Enter the key: "))  
code = input("Please enter encrypted text: ")  
  
for character in code:  
    if character in alphabet:  
        position = alphabet.find(character)  
        newPosition = (position - key) % 26  
        newCharacter = alphabet[newPosition]  
        restoredMessage += newCharacter  
    else:  
        restoredMessage += character  
  
print("The original message is: " + restoredMessage)
```

Make a new variable called `restoredMessage` to add the characters to

The only difference in the code is that when declaring `newPosition`, do the `position-key` instead of adding it, otherwise is the same entirely

# Conclusion

- Learning outcomes:
  - ✓ You should be able to confidently declare variables with two different data types (integer and string)
  - ✓ Use input function to ask the user to enter the value of a variable
  - ✓ Use a for loop
  - ✓ Use an if statement
  - ✓ Be able to use two string methods (accessing strings and find() function)
  - ✓ Know basic and some advanced arithmetic operators to calculate
  - ✓ Understand and be able to explain how a Caesar cipher works and the importance of the cipher





# Links to everyday life...

At work- Play- Helping others

play - can be used to  
create secret messages  
between friends

at work-  
cybersecurity and  
encrypting  
messages

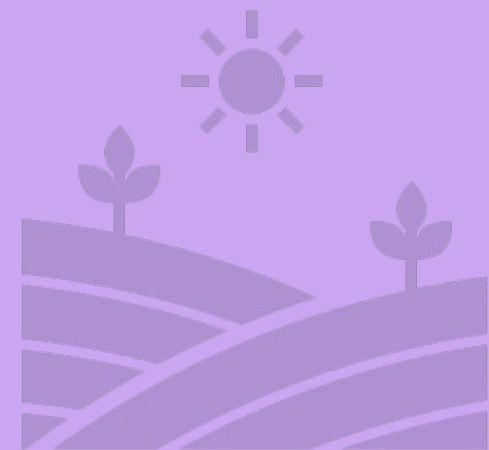
helping others-  
used by Julius  
Caesar to win wars  
and revolutionise  
Rome

More about Julius Caesar:

□ <https://www.theschoolrun.com/homework-help/Julius-caesar>

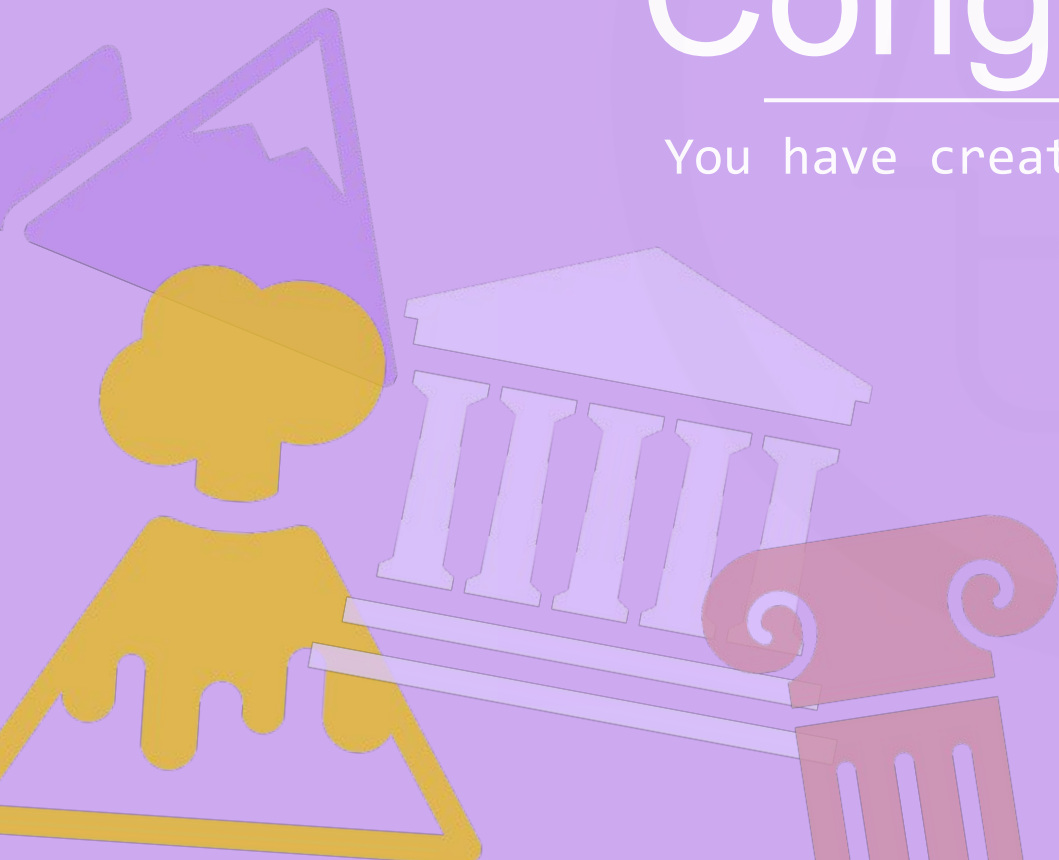
More about how to do the Caesar Cipher:

<https://projects.raspberrypi.org/en/projects/secret-messages/1>



# Congratulations!

You have created a Caesar cipher in Python



{4}

Python & Java 4 Teachers